

Benchmarks

Open Questions and DOL Benchmarks

Iuliana Bacivarov
ETH Zürich





Outline

- Benchmarks – what do we need?
 - what is available?
- Provided benchmarks in a DOL format
- Open questions

Benchmarks



How to assess the performance of an MPSoC design?

How to compare different MPSoC designs?

→ *Benchmarks are needed!*

- We need application benchmarks which are
 - Platform independent
 - Close to real applications
 - Provide the source code (or task graphs & performance data)
 - Freely available
- ... and for a comparison basis we need a common platform



→ *Goal: set up an MPSoC benchmark infrastructure and provide a simple interface*

Some Existing Benchmark Suites

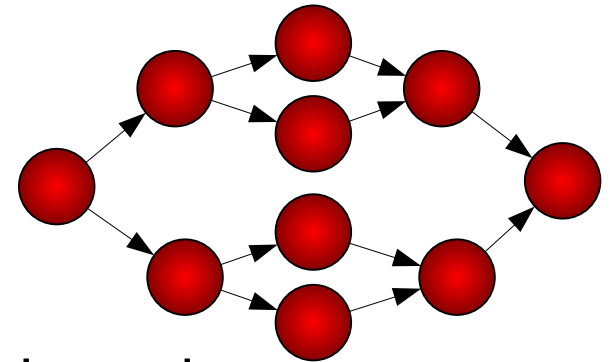
There are some available benchmark suites, but no open initiative for MPSoC (KPN executing on MPSoC)

- EMBC – Embedded Microprocessor Benchmark Consortium
<http://www.eembc.org> – license is required...
- E3S – Embedded System Synthesis Benchmarks Suite
<http://ziyang.eecs.northwestern.edu/~dickrp/e3s> – no access to the source code of benchmarks...
- PARSEC – Princeton Application Repository for Shared-Memory Computers <http://parsec.cs.princeton.edu>
- StreamIt
<http://www.cag.lcs.mit.edu/streamit/shtml/benchmarks.shtml>
– benchmarks in StreamIt language (for SDF applications)...
- ... and others

Applications

■ *Which type of benchmarks would be more useful?*

- ☐ Realistic vs. synthetic
- ☐ Large vs. small
- ☐ Scalability
- ☐ Granularity
- ☐ Computation vs. communication dominated



- ## ■ Need of a portable and simple specification format
- ☐ Architecture-independent API (e.g. YAPI)
 - ☐ Simple specification language (e.g. XML and C/C++ used in DOL, ESPAM, Artemis, MAMPS)
- ## ■ Features: granularity, scalability, and diversity

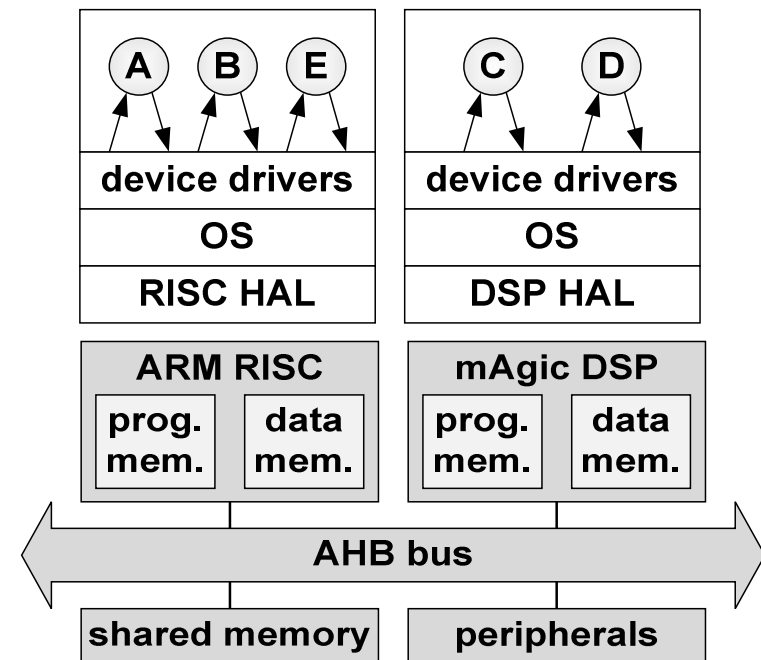
Platforms

Platforms

- Industrial/commercial (e.g. Cell BE) vs. academic/ad-hoc platform
- Virtual platform vs. real board

Run-time environment

- E.g. Linux, RTEMS, custom, ...
- Interesting features
 - Different schedules
 - Monitoring capabilities
 - Clean (only required services)



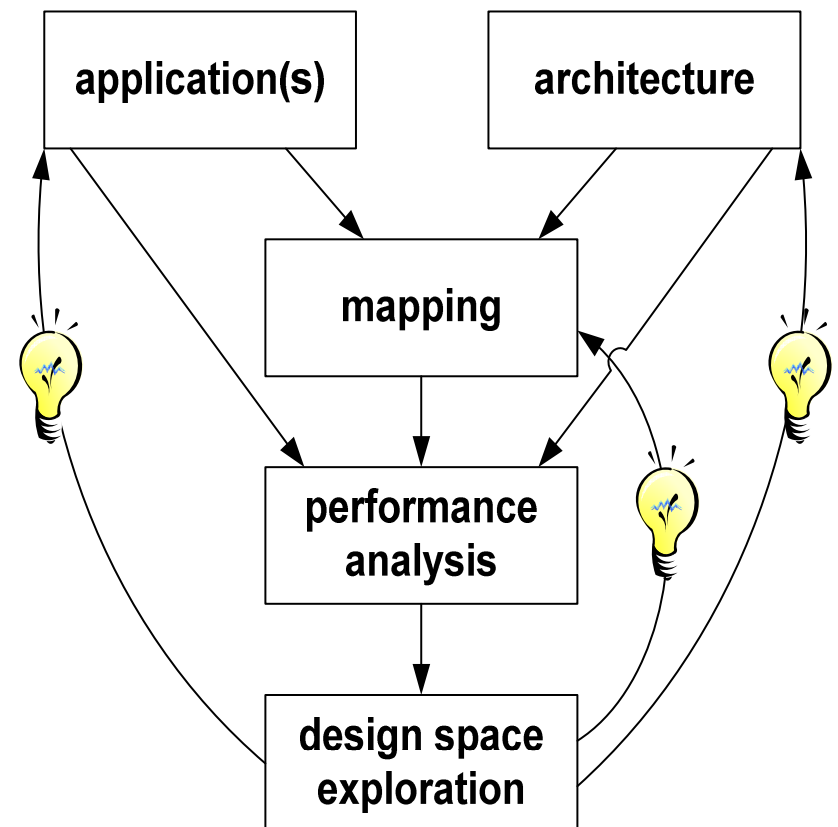
Expected features

- Speed vs. accuracy
- Observability, monitoring
- Possibility to extend (modify)
- Ease of use, documentation
- Compatibility with design goals

Application of Benchmarks

Evaluate steps in a design flow

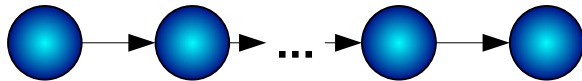
- Specification
- Synthesis
- Performance analysis (system-level)
 - Requires monitoring capabilities
 - ... can be done even with synthetic benchmarks
 - this requires pre-characterization
 - ... different for every framework
- Design space exploration
 - Requires performance monitoring



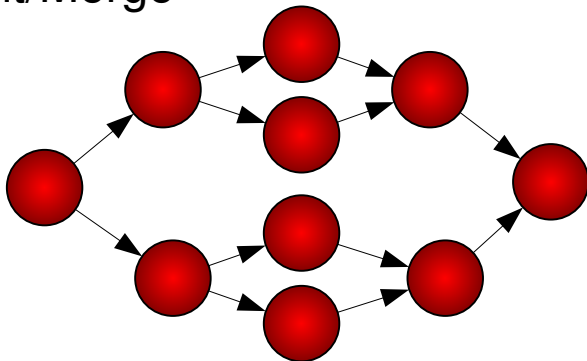
Synthetic Benchmarks

■ Different scenarios

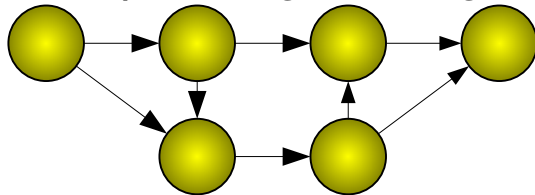
Pipeline



Split/Merge



More complex, e.g. filtering, FFT, etc.



■ Parameterization

- Computation or communication dominated
- Scalability (number of processes)
- Configurations

■ Need of performance data for design

- System-level performance analysis
- Design space exploration

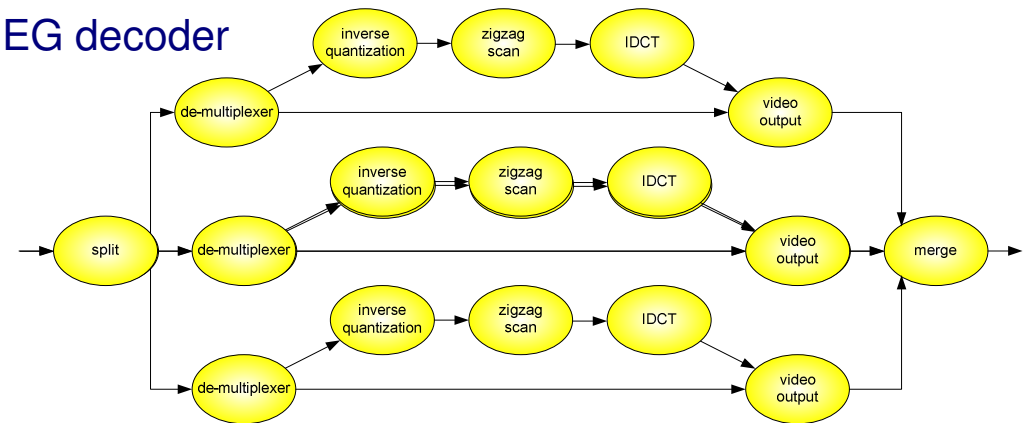
DOL Application Benchmarks

- “Larger”/“realistic” benchmarks MPEG-2 and MJPEG decoders on-line
<http://www.artist-embedded.org/artist/Benchmarks.html>

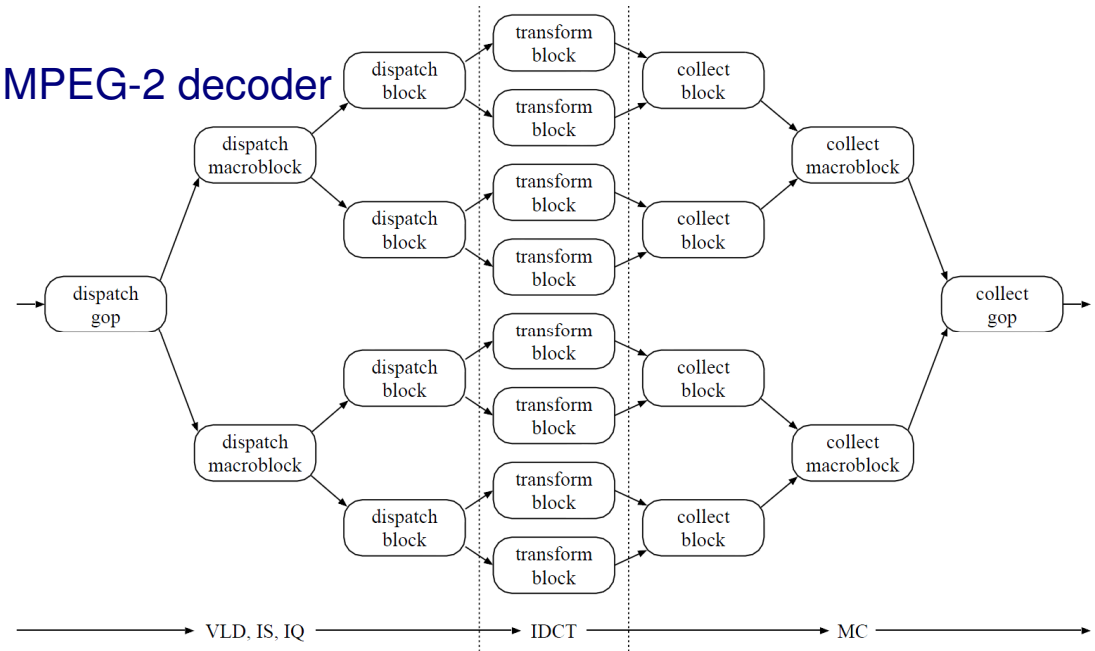
- Parallel DOL description
- Scalable

- ... “smaller” benchmarks, e.g. producer-consumer, FFT, etc., in the DOL package
<http://www.tik.ee.ethz.ch/~shapes>

MJPEG decoder



MPEG-2 decoder

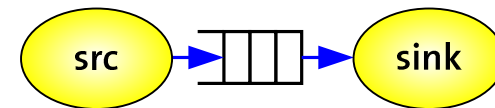


DOL Programming Model

```

01: <process name="src">
02:   <port type="output" name="out"/>
03:   <source type="c" location="src.c"/>
04: </process>
05:
06: <process name="sink">
07:   <port type="input" name="in"/>
08:   <source type="c" location="sink.c"/>
09: </process>
10:
11: <connection name="con_src_fifo">
12:   <origin name="src"><port name="out"/></origin>
13:   <target name="sink"><port name="in"/></target>
14: </connection>

```



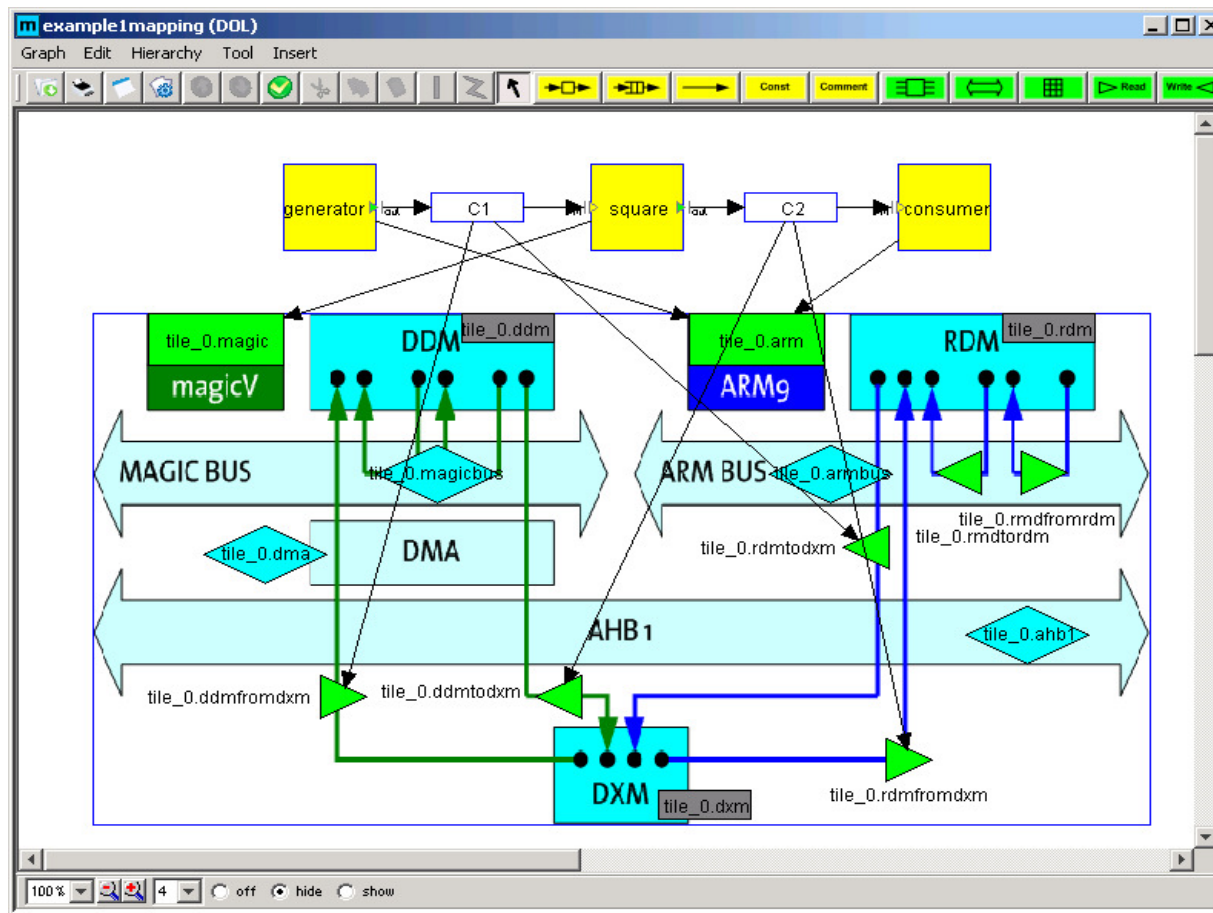
```

01: procedure INIT(Process p)
02:   //initialize process state
03: end procedure
04:
05: procedure FIRE(Process p)
06:   DOL_READ(input, size, buf)
07:   //processing
08:   DOL_WRITE(output, size, buf)
09: end procedure

```

- **Model of computation:** Kahn process network
 - Coordination language: XML
 - Functionality of processes: C/C++ with specific programming DOL API
- **Scalability:** “iterators” for large, multi-tile descriptions

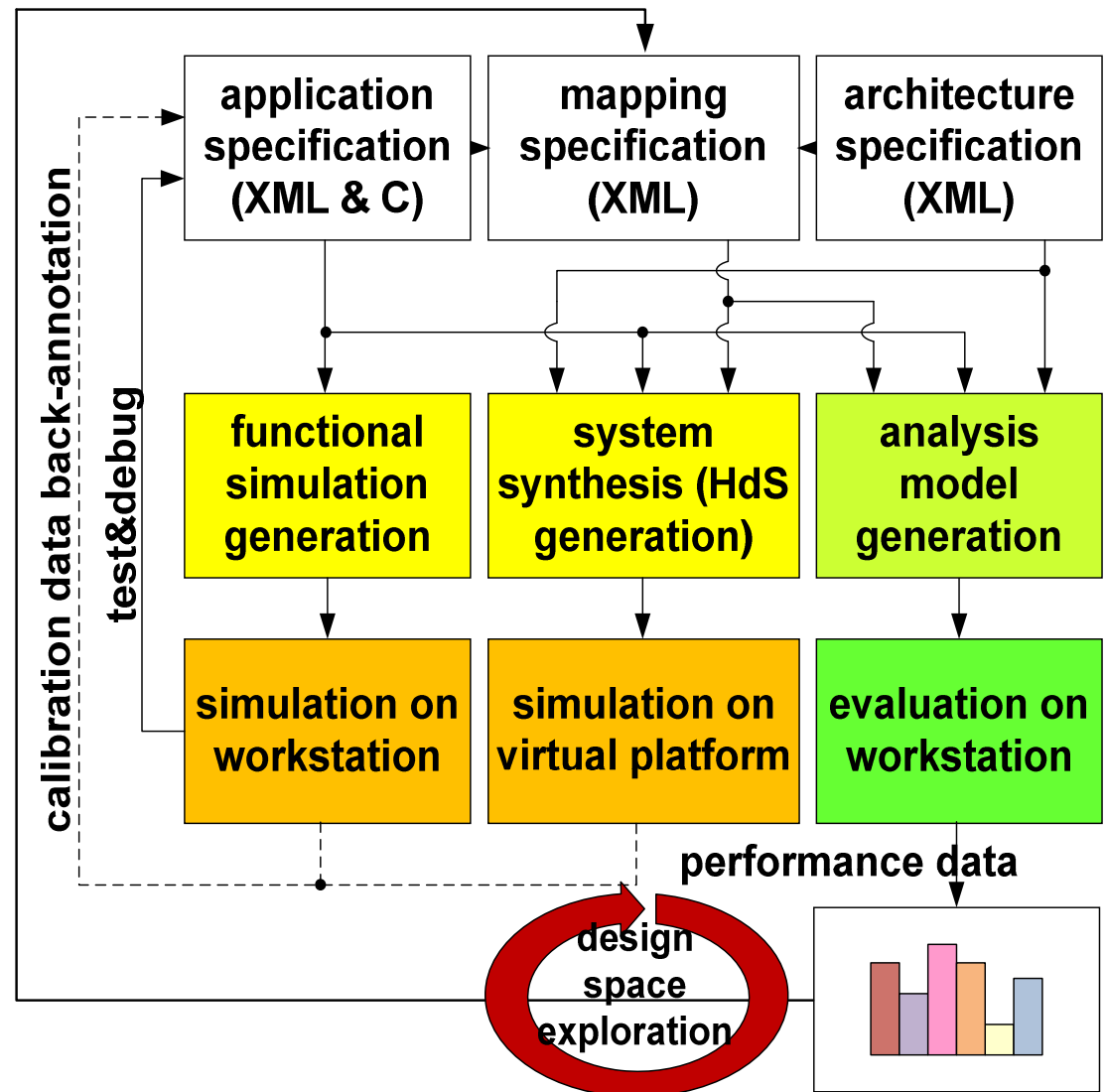
DOL Specification GUI



- *Graphical specification* of process networks, architectures, and mappings
- *Creation of DOL-compliant XML files* from the graphical specification
- *Generation of DOL-compliant C code templates* for processes

DOL Programming Environment

- *Scalable specification:*
application,
architecture, mapping
- *Functional simulation:*
debugging and profiling
(SystemC)
- *Synthesis back-ends:*
MPARM, Cell BE,
ATMEL Diopsis
- *Design space exploration:*
performance analysis
and optimization



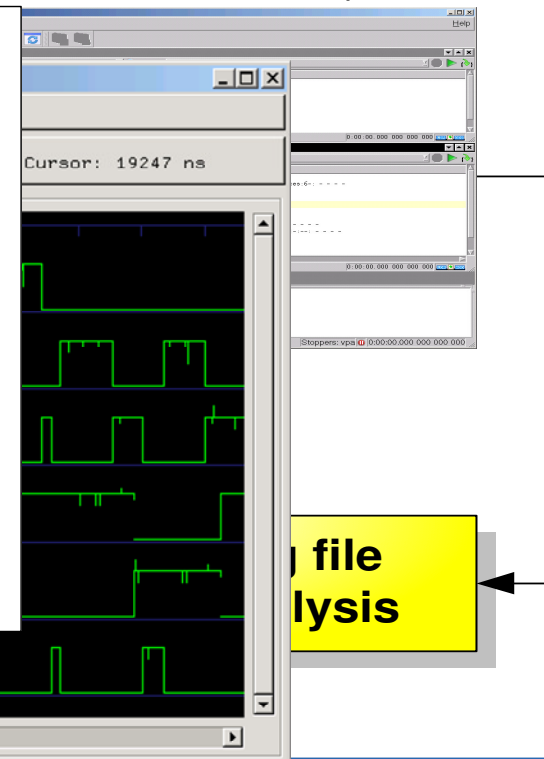
Performance Monitoring

- **Goal:** *automatically generate and calibrate the formal performance analysis model for design space exploration*
- **Automatic monitoring**
 - Automatic extraction of performance data
 - Automatic generation of simulation traces (for visualization)

```

01: <process name="control" basename="control" range="">
02:   <port name="out_scale" basename="out_scale" range="" type="output"/>
03:   <port name="out_delay" basename="out_delay" range="" type="output"/>
04:   <source type="c" location="control.c"/>
05:   <profiling name="NumOfFires" value="34000"/>
06:   <profiling name="WCED arm" value="16480000"/>
07:   <profiling name="BCED arm" value="11770000"/>
08:   <profiling name="WCED arm out_scale d940_extfifo" value="149900000"/>
09:   <profiling name="BCED arm out_scale d940_extfifo" value="140880000"/>
10:   <profiling name="WCED arm out_delay d940_extfifo" value="130810000"/>
11:   <profiling name="BCED arm out_delay d940_extfifo" value="129360000"/>
12:   <profiling name="WCED dsp" value="5080000"/>
13:   <profiling name="BCED dsp" value="4760000"/>
14:   <profiling name="WCED dsp out_scale d940_extfifo" value="343430000"/>
15:   <profiling name="BCED dsp out_scale d940_extfifo" value="81370000"/>
16:   <profiling name="WCED dsp out_delay d940_extfifo" value="235870000"/>
17:   <profiling name="BCED dsp out_delay d940_extfifo" value="67050000"/>
18: </process>

```





DOL Required Performance Data

(Static) Performance Data

- BCET/WCET of tasks (on different resources)
- Runtime overhead of OS
- Workload data: resource demand (CPU-cycles)
(e.g. needed to process 1, 2, 3, ... macroblocks)
- Characterization of system inputs as arrival curves
- Static power consumption of resources

(Dynamic) Performance Data

- Extract traces with timing information from any point in the system for the construction of arrival curves
- System-level performance data: CPU utilization, buffer requirements, end-to-end delay
- Dynamic power consumption



Applications, Architecture, Infrastructure

■ Applications

- Library of (“well-known” realistic/synthetic) distributed applications, easy to adapt and execute in a “reasonable” time
- Portable, simple specification format
- Features: granularity, scalability, and diversity

■ Architecture

- Easily change the configuration of the platform (and simulator): CPU speed, scheduling, number of CPUs
- Facilities to monitor the execution: e.g. scripts

■ Infrastructure

- Environment with all libraries, compilers, linkers, simulators, applications (set-up and able to run)

... Some Open Issues

- What benchmark set is useful?
- What would be a “suitable” specification format?
- What would be a “suitable” architecture?

- How to reach our goal?



→ Goal: set up an MPSoC benchmark infrastructure (set of applications ... and a basis platform) and provide a simple interface